REMARKS

This response respectfully traverses the objections and rejections in the office Action dated 12/28/2006.

Claims 1-20 are in the case.

Claims 19-20 were rejected under 35 U.S.C. 112, second paragraph, citing a lack of antecedent for the term "the method." The antecedent has been provided herein.

The Examiner rejected claim 1- 3, and 5-20 under 35 U.S.C. 103(a) as being unpatentable over Lin et al. (Lin) (U.S. Pub. No. 20020073211) in view of U.S. patent nop. 6886035 to Wolff (Wolff).

As presently amended, claim 1 (and all the other independent claims with corresponding wordings) contains the following limitation:

> an access monitor coupled to the interface, wherein the access monitor ***passively*** recognizes and tracks memory cycles ***by snooping memory address and control lines*** associated with the at least one memory block during a specified period of time and collects statistics associated with the session

No new matter is added, please refer to page 8, lines 1-8, of the original application as filed. "Snoop" is a term of art known to those skilled in the art generally meaning monitor, receive or listen the signals on the lines between two or more other entities without those entities knowing that you have received them. The "snooping" does not alter, change or affect the signals in any way.

Neither Lin nor Wolff describe or suggest this passive snooping of the address and control lines of a memory block.

On page 3, the Examiner points to Lin, FIG. 3, paragraphs 46, 47, and 65 as disclosing an "access monitor." Lin's paragraph 47 is instructive in this matter. In this paragraph it is stated, "...Once a session is initiated, a state server monitoring thread is created between the webserver and the state server to monitor and track the session occurring with the user. A **monitoring thread** as used here is software code application that, **when executed** by a processor, creates a mechanism for facilitating communication between the state server and the webserver so that the state server can **send and receive signals** to and from the webserver **to monitor** its activities..." If a connection terminates, ".. The state server will have retained relevant information session information so that an application server may attempt to reconnect and possibly get rerouted to another webserver to continue a session."

A fair reading of the above passage is that he monitor is active software that interacts (sending and receiving signals) with the system modules. A "thread" is active software. Lin says that the "thread" is executed and facilitates sending and receiving signals ... to monitor. The monitoring is by sending and receiving signals, Lin is not describing the reporting of results but the actual monitoring. In contrast the present invention, as now claimed, passively monitors by snooping the signals. There is no sending component of the present invention's monitoring. The present invention only sends to report results.

Moreover, Lin, in order to restore a connection that was terminated, must, by definition "know" more that just the number of reads or writes to a memory. Lin's monitor by definition "knows" much more in order to possible restore terminated operations. The present invention as claims passively monitors the signals to the memory and does not send anything in order to monitor the activity. Only later will the monitor report results. The present invention access monitor cannot restore or know how to restore a session.

Wolff, on the other hand, only monitors I/O operations, see column 11, line 27. Wolff, from his Abstract, can dynamically rebalance itself to optimize throughput by migrating client I/O requests from over_utilized pathways to under_utilized pathways.

Wolff is not and does not suggest passively monitoring memory control and address lines, he is balancing transfers equally over pathways.

The amendment adds that the monitoring is passive, and that monitoring is by snooping the signals on the memory control and address lines, but not interfering with those signals between the memories and the memory interfaces.

Respectfully the Examiner on page 4, item 10, mentions that Wolff, "recognizes memory cycles being executed on memory blocks." The Examiner cites Wolff's FIG. 14 and column 64, lines 8-47.. But, this excerpt is describing memory file systems with pointers, lengths, etc. Record lengths, overflow indicators, non-contiguous memory files are all handled. Moreover, such memory file systems are acknowledged as common in the field. However, the excerpt, to my reading, does not suggested passively detecting

and tracking memory cycles, much less tracking memory cycles to be used to reroute pathways in Wolff, and even less about tracking memory cycles to load balance as presently claimed in the instant application.

Claim 4 was rejected under 35 U.S.C. 1039a0 citing Lin, Wolff and Bass (U.S. patent no. 6449576). Item 14 on page 5 of the Office Action states that Bass teaches an access monitor embodied in an ASIC. But, since Claim 1 is distinguished, and claim 4 depends from claim1, Bass is moot. However, Bass does not add the passive monitoring of the memory address and control lines for load balancing.

Briefly reviewing the list of patents on page 6, of the Office Action, these patents do not appear to add the passive monitoring of the actual memory control and address lines as now claimed in all the independent claims.

It is respectfully held that all claims 1-20 in the present application are now allowable and a Notice of Allowance is requested.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

Edwin H. Paul

Edwin H. Paul
Reg. No. 31,405
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500